METHOD AND APPARATUS FOR POWER OPTIMIZATION DURING AN INTEGRATED CIRCUIT DESIGN PROCESS

FIELD OF THE INVENTION

[0001] One or more aspects of the present invention relate generally to computer-aided design of electronic circuits and, more particularly, to power optimization during an integrated circuit design process.

BACKGROUND OF THE INVENTION

[0002] Power consumption has become an important design challenge in modern integrated circuit (IC) devices, for example, programmable logic devices (PLDs) and standard cell application specific integrated circuits (ASICs). PLDs exist as a well-known type of IC that may be programmed by a user to perform specified logic functions. There are different types of programmable logic devices, such as programmable logic arrays (PLAs) and complex programmable logic devices (CPLDs). One type of programmable logic device, known as a field programmable gate array (FPGA), is very popular because of a superior combination of capacity, flexibility, time-to-market, and cost.

[0003] An FPGA typically includes an array of configurable logic blocks (CLBs) surrounded by a ring of programmable input/output blocks (IOBs). The CLBs and IOBs are interconnected by a programmable interconnect structure (routing resources). The CLBs, IOBs, and interconnect structure are typically programmed by loading a stream of configuration data (bitstream) into internal configuration memory cells that define how the CLBs, IOBs, and interconnect structure are configured. The configuration bitstream may be read from an external memory, conventionally an external integrated circuit memory EEPROM, EPROM, PROM, and the like, though other types of memory may be used. The collective states of the individual memory cells then determine the function of the FPGA.

[0004] FIG. 1 is a flow diagram depicting a conventional design process ("CAD process 100") for an integrated circuit, such as an ASIC or an FPGA. CAD process 100 begins by receiving a design 101. Design 101 represents a circuit design at a schematic or logic level. At step 102, design 101 is synthesized to produce a logical network list ("netlist") supported by the target device. At step 103, the synthesized design is mapped onto primitive components within the target device (e.g., programmable logic blocks of an FPGA). At step 104, placement for the components of the synthesized and mapped design is determined for the target device. At step 105, interconnects (e.g., signal conductors) are routed within the target device for the placed components. CAD process 100 ends by producing a CAD result 106. Steps 101 through 105 are referred to herein as standard map-place-route (SMPR) process 110.

[0005] Performance and frequency requirements translate to a maximum allowable delay for paths traversing routing resources between circuit elements. As such, the conventional design process is "timing-driven" in that SMPR process 110 is optimized in accordance with timing constraints. SMPR process 110, however, does not optimize design 101 for power. Typically, incorporating power optimization in the conventional design process adversely affects the timing performance of the design.

[0006] Accordingly, there exists a need in the art for an integrated circuit design process that optimizes power without adversely affecting the timing performance of the design.

SUMMARY OF THE INVENTION

[0007] An aspect of the invention relates to designing an integrated circuit having a plurality of logic paths. The integrated circuit is designed in accordance with timing constraint data. Any logic paths in the plurality of logic paths that have a timing characteristic within a threshold are identified and define a first set of logic paths. Any logic paths in the plurality of logic paths other than those in the first set of logic paths define a second set of logic paths. The integrated circuit is then selectively optimized to reduce power consumption in response to the first set of logic paths and the second set of logic paths.

[0008] Another aspect of the invention relates to designing an integrated circuit. The integrated circuit is first designed in accordance with timing constraint data. Timing critical logic circuitry is then identified. For example, the timing critical logic circuitry may be defined with respect to timing constraint data. The integrated circuit is then selectively optimized in response to the timing critical circuitry. In an embodiment, logic circuitry other than said timing critical circuitry is power optimized. In another embodiment, the integrated circuit is power optimized and a determination is made as to whether a timing characteristic of the timing critical circuitry has been modified beyond a threshold.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] Accompanying drawing(s) show exemplary embodiment(s) in accordance with one or more aspects of the invention; however, the accompanying drawing(s) should not be taken to limit the invention to the embodiment(s) shown, but are for explanation and understanding only.

[0010] FIG. 1 is a flow diagram depicting a conventional design process for an integrated circuit;

[0011] FIG. 2 is a flow diagram depicting an exemplary embodiment of an integrated circuit design process incorporating power optimization;

[0012] FIG. 3 is a flow diagram depicting another exemplary embodiment of an integrated circuit design process logic circuitry ("logic block") and the connections therein. In this respect, design 101 may include a multiplicity of logic paths. Timing constraints, for example, may include frequency requirements and maximum logic path delay for a particular target device, such as a field programmable gate array (FPGA).

[0017] Notably, FIG. 5 depicts a block diagram of an exemplary embodiment of an FPGA 500 coupled to a program memory 512. FPGA 500 illustratively includes CLBs 507, I/O routing ring 506A ("programmable interconnect"), memory 511, delay lock loop (DLL) blocks 509, multiply/divide/de-skew clock circuits 510, and programmable IOBs 506B. DLL blocks 509 and clock circuits 510 collectively provide digital clock management (DCM) circuits for managing clock signals within FPGA 500. FPGA 500 may include other types of logic blocks and circuits in addition to those described herein.

[0018] CLBs 507 are programmably connectable to each other, and to I/O routing ring 508, for performing various types of logic functions. Each of CLBs 507 may include one or more "slices" and programmable interconnect circuitry (not shown). Each CLB slice in turn includes various circuits, such as flip-flops, function generators (e.g., look-up tables (LUTs)), logic gates, memory, and like type well-known circuits.

[0019] Programmable IOBs 506B are configured to provide input to, and receive output from, one or more of CLBs 507. Configuration information for CLBs 507, I/O routing ring 506A, and programmable IOBs 506B is stored in memory 511. Briefly stated, a configuration bitstream produced from program memory 512 is coupled to a configuration port 502 of FPGA 500 to implement a desired circuit therein. Each of CLBs 507, I/O routing ring 506A, and programmable IOBs 506B are generally referred to herein as "programmable logic blocks".

[0020] With renewed reference to FIG. 2, as described above, SMPR process 110 does not optimize design 101 for power. That is, SMPR process 110 does not attempt to reduce power consumption during the synthesis, map, place, and routing processes. At step 201, a determination is made as to whether power is to be optimized. If not, CAD process 200 outputs CAD result 106. If so, CAD process 200 proceeds to step 202. Steps 110 and 201 are referred to herein as SMPR process with power option 211. For users who are not concerned with power consumption, SMPR process with power option 211 allows the power optimization to be skipped such that the run-time of CAD process 200 is not increased.

[0021] At step 202, critical logic paths in design 101 are identified. A "critical logic path" is a logic path within design 101 that has a timing characteristic that falls within a particular threshold. The threshold may be pre-defined or may be determined by the user. In an embodiment, the threshold is defined as a certain percentage of the timing constraint data used with SMPR process 110. For example, the timing constraint data may include a maximum logic path delay. Thus, a critical logic path may be a logic path exhibiting a delay that falls within a pre-defined or user-selectable percentage of the maximum logic path delay. The difference between the maximum logic path delay and the delay actually exhibited by a logic path is referred to as "timing slack". Thus, in one embodiment of the invention, logic paths having timing slack with less than x% of the maximum path delay are identified at step 202 as being critical logic paths. The parameter, x, may be a pre-defined value or may be established by the user. For example, the value of x% may be 10%, such that if the maximum path delay is 20 ns, then all logic paths with a delay of greater than 18 ns are deemed critical logic paths.

[0022] In another embodiment, the threshold may be defined in terms of an absolute delay value. For example, the timing constraint data may include a minimum logic path delay.

incorporating power optimization;

[0013] FIG. 4 is a block diagram depicting an exemplary embodiment of a computer system suitable for implementing processes and methods described herein; and

[0014] FIG. 5 is a block diagram depicting an exemplary embodiment of a field programmable gate array (FPGA) coupled to a program memory.

DETAILED DESCRIPTION OF THE DRAWINGS

[0015] Method and apparatus for power optimization during an integrated circuit design process is described. One or more aspects in accordance with the invention are described in terms of a field programmable gate array (FPGA). While specific reference is made to an FPGA, those skilled in the art will appreciate that one or more aspects of the invention may be used for designing other types of integrated circuits, such as complex programmable logic devices (CPLDs), application specific integrated circuits (ASICs), and the like.

[0016] FIG. 2 is a flow diagram depicting an exemplary embodiment of a CAD process 200 for an integrated circuit that incorporates power optimization. CAD process 200 receives design 101. CAD process 200 then executes SMPR process 110 described above with respect to FIG. 1. Notably, SMPR process 110 will produce CAD result 106 in accordance with established timing constraint data. Timing constraint data includes a series of constraints applied to logic pathways ("logic paths") within design 101 that dictate the desired performance of the design in the time domain. Logic paths of design 101 include logical circuit configurations and connections or "nets" for connecting logical circuit configurations. For example, a logic path may be a series of logic components having a source and a sink and the signal paths that connect the logic components. In another example, a logic path may be an entire logic subsystem or block of

Thus, the critical logic paths may comprise all logic paths that are less than, a predetermined number, e.g., 5 ns slower than the minimum path delay. Those skilled in the art will appreciate that various types of absolute and relative thresholds may be used to determine critical logic paths (also referred to as timing critical logic circuitry).

[0023] The critical logic paths identified in step 202 are locked at step 203. That is, the critical logic paths will not be optimized for power. At step 204, design 101 is synthesized to produce a netlist supported by the target device while attempting to reduce power consumption ("power-driven synthesis"). Power-driven synthesis may be performed in accordance with power constraints entered by the user, or may attempt to save the most power possible. For example, power reduction may be achieved using finite state machine (FSM) synthesis (e.g., state encoding, partitioning), arithmetic block synthesis (e.g., choice of adder), or by shielding inputs to functional blocks and/or balancing logic trees to reduce signal glitches.

[0024] At step 205, the synthesized design is mapped onto primitive components within the target device (e.g., programmable logic blocks) in accordance with the power constraints ("power-driven mapping"). For example, power-driven mapping may focus on maintaining high-activity nets within logic cells, or performing logic replication carefully and sparingly to avoid an increase in power.

[0025] At step 206, components of the synthesized and mapped design are placed within the target device ("power-driven placement"). For example, power-driven placement may focus on minimizing the distance of logic blocks whose interconnecting nets have high switching activity.

[0026] At step 207, interconnects are routed within the target device for the placed components ("power-driven routing"). For example, power-driven routing may attempt to reduce power consumption by minimizing the lengths of highfan-out and high-activity nets.

[0027] At optional step 209, a determination is made as to whether any of the power optimized logic paths now violate the timing constraint data. If so, CAD process 200 proceeds to optional step 212, where the offending power optimized logic paths are rejected and/or reported to the user. CAD process 200 ends by producing a CAD result 208. Steps 204, 205, 206, 207, and optional steps 209 and 212 are referred to herein as power SMPR process 210. The critical logic paths that were locked in step 203 are not optimized for power during power SMPR process 210. While specific processes for reducing power consumption during the synthesis, map, place, and route phases were described above, those skilled in the art will appreciate that various other power-saving processes may be employed that are known in the art.

[0028] FIG. 3 is a flow diagram depicting another exemplary embodiment of a CAD process 300 for an integrated circuit that incorporates power optimization. Steps that are the same or similar to those shown in FIG. 2 are designated with identical reference numerals and are described in detail above. In the present embodiment, process 300 proceeds from SMPR process with power option 211 to step 203 without identifying or locking the critical logic paths. Process 300 then executes the SMPR process 201, and proceeds to step 302. At step 302, a determination is made as to whether a power optimization for any of the critical logic paths has adversely affected the timing characteristic beyond a pre-defined or user selectable threshold. If so, the offending power optimizations for the critical logic paths are rejected or reported to the user at step 304. For example, if for a given critical logic path a power optimization results in the decrease of timing slack beyond a threshold, the power optimization may be rejected or reported to the user. Alternatively, if for a given critical logic path a power optimization results in an increase in delay beyond an absolute delay value, or a delay value relative to the timing constraint data, the power optimization may be rejected or reported to the user.

[0029] FIG. 4 depicts a block diagram showing an exemplary embodiment of a computer 400 suitable for implementing processes and methods described below. Computer 400 includes a central processing unit (CPU) 401, a memory 403, and an I/O interface 402. CPU 401 may be any type of microprocessor known in the art. I/O interface 402 may be directly coupled to memory 403 or coupled through CPU 401. I/O interface 402 may be coupled to various input devices 412 and output devices 411, such as a conventional keyboard, mouse, printer, display, and the like. In addition, I/O interface 402 may be coupled to a network 413 and may be adapted for communication with a database 414 to receive and transmit data, such as data files and the like. Notably, database 414 may store design 101 and CAD result 208, described above with respect to FIG. 2.

[0030] Memory 403 may store all or portions of one or more programs and/or data to implement the processes and methods described above. Although the invention is disclosed as being implemented as a computer executing a software program, those skilled in the art will appreciate that the invention may be implemented in hardware, software, or a combination of hardware and software. Such implementations may include a number of processors independently executing various programs and dedicated hardware, such as application specific integrated circuits (ASICs).

[0031] Computer 400 may be programmed with an operating system, which may be OS/2, Java Virtual Machine, Linux, Solaris, Unix, Windows, Windows95, Windows98, Windows NT, and Windows2000, WindowsME, and WindowsXP, among other known platforms. At least a portion of an operating system may be disposed in memory 403. Memory 403 may include one or more of the following random access memory, read only memory, magneto-resistive read/write memory, optical read/write memory, cache memory, magnetic read/write memory, and the like, as well as signal-bearing media as described below. Memory 403 may store all or a portion of process 200 or process 300, described above with respect to FIGs. 2 and 3.

[0032] An aspect of the invention is implemented as a program product for use with a computer system. Program(s) of the program product defines functions of embodiments and can be contained on a variety of signal-bearing media, which include, but are not limited to: (i) information permanently stored on non-writable storage media (*e.g.*, read-only memory devices within a computer such as CD-ROM or DVD-ROM disks readable by a CD-ROM drive or a DVD drive); (ii) alterable information stored on writable storage media (*e.g.*, floppy disks within a diskette drive or hard-disk drive or read/writable CD or read/writable DVD); or (iii) information conveyed to a computer by a communications medium, such as through a computer or telephone network, including wireless communications. The latter embodiment specifically includes information downloaded from the Internet and other networks. Such signal-bearing media, when carrying computer-readable instructions that direct functions of the invention, represent embodiments of the invention.

[0033] Method and apparatus for power optimization during integrated circuit design process has been described. The power optimization design process achieves substantial power reduction when compared with the conventional design process. The power optimization design process maintains the timing performance of the design's critical signal paths, while optimizing the power consumption of the non-critical signal paths.

[0034] While the foregoing describes exemplary embodiment(s) in accordance with one or more aspects of the present invention, other and further embodiment(s) in accordance with the one or more aspects of the present invention may be devised without departing from the scope thereof, which is determined by the claim(s) that follow and equivalents thereof. Claim(s) listing steps do not imply any order of the steps. Trademarks are the property of their respective owners.